

平成 23 年度 情報工学コース卒業研究報告要旨

坂部 研究室	氏 名	近 藤 隆
卒業研究題目	SAT ソルバを利用した充足可能性判定を支援する SML ライブラリの開発	
<p>論理式の充足可能性問題 (SAT 問題) は NP 完全であり, 計算に大変時間が掛かることが知られている. この問題を短時間で解くことを目指し, 最近では SAT ソルバと呼ばれるツールの開発が競って進められている. しかし, SAT ソルバを利用するためには解きたい問題を CNF 論理式へと変換しなければならず, また, 充足可能性判定後には, 充足可能性の判定結果から本来の問題の解へと復元する必要があるなど, 手間の掛かるものが多い.</p> <p>そこで, 本研究では, SAT ソルバを利用する上で利便性が向上し, かつ, 時間的に効率良く SAT ソルバを用いて解きたい問題を解くことのできるライブラリの開発を行う. そのためには, SAT ソルバを用いて解きたい問題から充足可能性判定用の論理式へと変換し, その後その論理式を CNF 論理式に変換する必要があるが, 前者の変換は自動化できないため, 各種ブール関数を用意することによりユーザへの支援を行うこととし, 後者の変換に対しては, Tseitin 変換と呼ばれる論理式を充足可能性的に等価な CNF 論理式に変換する手法を用いて自動で行うようにする. それと同時に, SAT ソルバからの充足可能性判定結果から本来の問題の解への復元を自動で行うライブラリの開発を行う必要もある.</p> <p>しかし, SAT ソルバを用いて解きたい問題には様々な形式が存在し, それらの問題に対してライブラリの汎用性を高めるため重要となる問題が 3 つ考えられる. 1 つ目の問題は, 問題ごとに必要とされる命題変数の型が異なるという問題であり, 2 つ目は, 問題ごとに入力する形式が異なるという問題, そして 3 つ目は, SAT ソルバの判定結果を用いて, 命題変数ではなく, 問題本来の解を出力をする際の出力形式が問題ごとに異なるという問題である.</p> <p>今回作成するライブラリにおいて, 1 つ目の問題に対しては, ファンクタと呼ばれる機能を用いる. 具体的に言うと, 命題変数の型のパラメータを引数に取るファンクタを用いるということである. これによって, 命題変数の型をユーザが問題ごとに変更することにより, あらゆる問題に対応することができる. また, 2 つ目の問題に対しては, DIMACS 形式での入力に対してはプログラム内でユーザが利用可能な形式に落としこみ, そこからの処理をユーザに任せるといった形で支援する機能を与える. そして 3 つ目の問題に対しては, 汎用的な実装に無理があるためライブラリでは対応を行わず, その問題から作られた充足可能性判定用論理式が, UNSAT のときは UNSAT という返り値を返し, SAT のときは, 充足可能性判定の結果から真となるリテラルのみ命題変数へと復元した返り値を返すこととする.</p> <p>また, 各種ブール関数を用意することにより, 問題から論理式への変換を支援しているが, 高々 1 つが真に解釈されるときのみ真に解釈されるブール関数 <code>Atmost1</code> や, 唯 1 つが真に解釈されるときのみ真に解釈されるブール関数 <code>Unique</code> などは CNF 論理式へ変換する方法が複数存在し, また, 変換した結果得られる CNF 論理式は変換方法によって異なっている. そこで, SAT ソルバを用いてその変換した CNF 論理式の実行時間の比較を行ったところ, 実行時間に差があることが確認できる. そのため, 充足可能性的には等価だが, 出力する CNF 論理式の異なる関数を複数個作成し, 比較実験を行い時間的に効率の良い関数をライブラリに組み込む.</p>		